Programming of parallel computers Computer Lab no. 4: Communicators and virtual topologies

DCAMM, DTU and IT, Uppsala University

January, 2008

The available test codes for this computer lab can be downloaded from
`http://www2.imm.dtu.dk/courses/FortranMPI/MPI/Labs/Lab4`.

Exercise 1 (Creating communicators)
You have at your disposal the routines `Communicator.f`, `communicators.*` and the subroutine `sub_comm_split.f` to see examples of creating communicators and using them.

Try to write your own code, which can perform the following exchange operation:
Assume that you have p processors and a matrix of dimension $p \times N$, which is in the memory of P0.
Tasks:

1. Send one row of the matrix to each of the processors.
2. Arrange so that the processors exchange their rows pairwise (P0 with P1, P2 with P3 etc). Assume that we have even number of those. For the latter, try to create suitable communicators.

Exercise 2 (Experience with a virtual Cartesian grid)

Create a periodic 2D Cartesian mesh virtual topology and perform one vertical and one horizontal shift communication in the two possible directions. Print the results for a small matrix (say 4×4 distributed on 4 PEs) to illustrate that the code is functioning correctly.

Exercise 3 (Computing matrix norms in parallel)
Assume that you have a matrix A of size $pN \times pN$ which has been already distributed blockwise among p processors, arranged in a 2D mesh (i.e.,$p = 1, 4, 9$ etc).
Compute in parallel the following matrix norms:

1. $\|A\|_F = \sqrt{\sum_{i,j=1}^{pN} A(i,j)}$, the so-called Frobenius norm.

2. $\|A\|_\infty$, which is computed as the maximum row sum of $|A|$.

Exercise 4 (Parallel search algorithm)
Study the code `search.f`. Compile it and run it on different number of processors. Does it perform correctly (according to the comments in the source)?
Modify it or write your own code in order to find the first zero in a list of N numbers, which are stored as 1D array in a distributed matter.

Exercise 5 (Parallel prefix)
Assume that you have an array A of total length pN and it is already equally distributed on p processors. Assume that it is initialized as random (or in any other way you may decide).
The task is to compute the global sum as well as all partial sums, namely,

$$S_k = \sum_{i=1}^{k} A(i),\ k = 1, 2, 3, \cdots, N$$

The array S has to be collected in one of the processors.